UNIVERSITÉ DE LORRAINE

L'INSTITUT DES SCIENCES DU DIGITAL MANAGEMENT COGNITION
TERMINOLOGY

Develop a Term Identification System for a Specific Domain Final Report

---

Authors:
Shalini PRIYA,
Soklay HENG

A final report for the Terminology class

Academic year 2021-2022

December 5, 2021

Contents

1 Introduction **1**

1.1 Team Organisation 1

1.2 Project Presentation 1

2 Methodology and Discussion on Results **1**

2.1 Rule-based system 1

2.2 Manual filtering of candidates 2

2.3 Fine-tuning the rule-based system 2

2.4 Generating annotated corpus in IOB format 3

2.5 Training a model 3

2.6 Evaluation 3

3 Challenges **5**

4 Conclusions **5**

1 Introduction

1.1 Team Organisation

We are a group of two, having different backgrounds - Computer Science and Linguistic. We have divided our tasks as follow: 1 of us is responsible for data collection, data processing, defining rule-based systems, and doing manual annotation; 1 of us is responsible for building an IOB-tagging model, training the model, and evaluating the model.

1.2 Project Presentation

Our project is putting forward keyword extraction method for domain-specific articles using a Spacy tagger component powered by statistical models which gives prediction based on the model's current weight values. We have collected 20 PDF articles in Computer Science domain for this term extraction project. Our project consists of following steps:

- Defining a rule-based system for extracting candidates for the first time
- Manual filtering of candidates
- Fine-tuning the rule-based system based on manually filtered candidates
- Generating annotated corpus in IOB format
- Training a model
- Applying the model on the new unseen article
- Evaluation

The details and results of each process will be illustrated in the following section.

2 Methodology and Discussion on Results

2.1 Rule-based system

We carried out term candidates extraction from the 20 PDF articles. We used *Textractor*[1] which is OCR application for converting all PDF articles to text in Python. The PDF's text contained junk characters, stop words, and URL which were removed during pre-processing. The cleaned text obtained was then segmented into tokens. A token is a lexeme that explicitly indicates its categorization for the purpose of parsing. These tokens represent the basic units of information in the articles and are subjected to further processing to generate the required training data. The rule

[1] https://github.com/skvark/Textractor

based-system was developed based on phrase pattern matching, such as NOUN, NOUN+NOUN, ADJ+NOUN, and NOUN+NOUN+NOUN.

Result : Following figure shows that 1880 words with its frequencies were filtered from 20 articles.

	terms	frequency
0	(N, model)	59
1	(N, table)	54
2	(N, language)	40
3	(N, words)	35
4	(N, generation)	32
...	...	...
1875	(N, summarization)	1
1876	(ADJ+N, geographic descriptions)	1
1877	(ADJ+N, Empirical methods)	1
1878	(N+N+N, language generation pages)	1
1879	(N+N, generation pages)	1

[1880 rows x 2 columns]

2.2 Manual filtering of candidates

We then manually filtered out the keywords; we did manual annotation by eliminating those who are not terms. The number of words which were terms were 84 in numbers.

2.3 Fine-tuning the rule-based system

We adapted the previous rule-based system according to the terms we got from manual annotation in order to get the exact number of keywords from the 20 articles. We were interested in exact word matching, including all inflected forms and different syntactic forms.

Result :From the below figure , it can be seen that we were able to filter 170 words from the 20 articles.

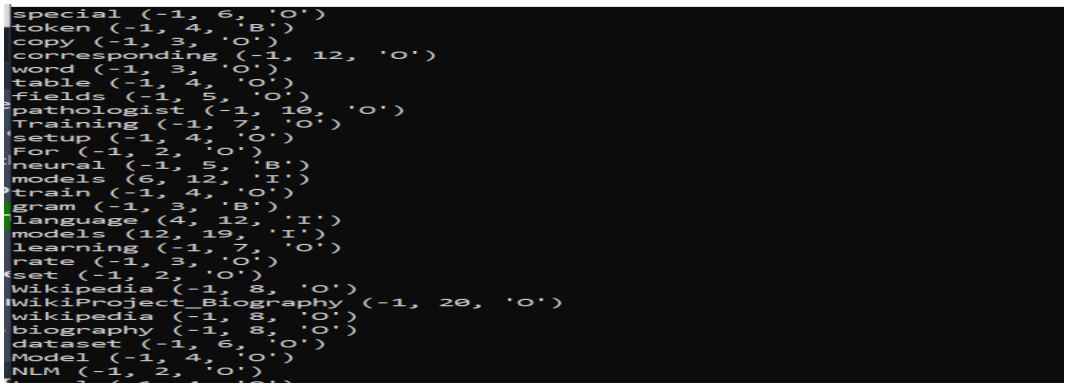
	terms	frequency
0	(rulebased, language model)	18
1	(rulebased, token)	12
2	(rulebased, language models)	10
3	(rulebased, vectors)	10
4	(rulebased, copy actions)	9
..	...	...
165	(rulebased, network conversation model)	1
166	(rulebased, string model)	1
167	(rulebased, forest string model)	1
168	(rulebased, cation machine translation)	1
169	(rulebased, machine translation summarization)	1

[170 rows x 2 columns]

2.4 Generating annotated corpus in IOB format

IOB annotation is applied on all the PDF texts with the rule-based system to automatically assign IOB tagging for each word in the corpora. A dataset is created with the extracted features of over 3000 words. To assign labels the PDF files has to be read, and its content must be comprehended. Each sequence of data is labelled as ”B” if it is a unigram, ”B” for starting and "I" for ending if it is bigram, and ”O” if it is not a keyword (not either in unigram or bigram). The model learns the weight of each feature and the dependencies between the tags.These annotated corpora were then used for training our IOB-tagging model.

Result: Each token is represented in IOB format as shown in the figure below.



2.5 Training a model

We are using *Spacy*[2] training model.

- We created a config file which contains all settings and hyperparameters for training pipeline. It defines the training process and pipeline
- pipelines used were ner, token2vector
- Created spacy files
    1. train.spacy - iob annotated corpus
    2. dev.spacy - unseen preprocessed corpus

2.6 Evaluation

We get two folders after training our models:

1. model-best : is the model that got the highest score on the dev set.
2. model-last: is the model trained in the last iteration. We used above models to evaluate our results for both train.spacy and dev.spacy (unseen article)

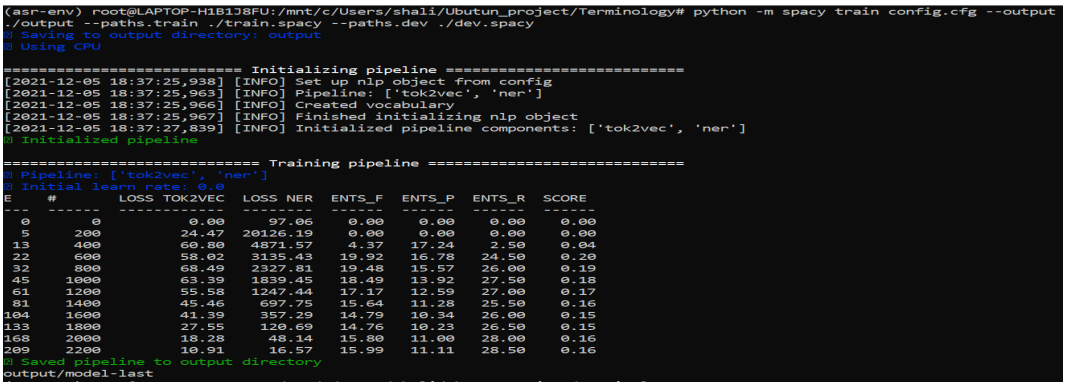
[2] https://spacy.io/

3. Ran comand line arguement to train the model: python -m spacy train config.cfg –output ./output –paths.train ./train –paths.dev ./dev

4. Ran comand line arguement to evaluate the model: python -m spacy evaluate model data path –output –code –gold-preproc –gpu-id –displacy-path –displacy-limit

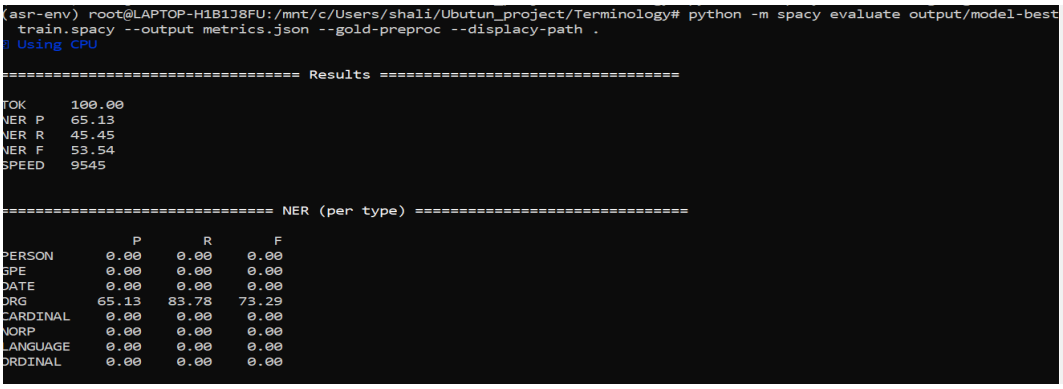
   Result: The model gives precision, recall and F-scores.

   (a) The model has been successfully trained.



```
(asr-env) root@LAPTOP-H1B1J8FU:/mnt/c/Users/shali/Ubutun_project/Terminology# python -m spacy train config.cfg --output
./output --paths.train ./train.spacy --paths.dev ./dev.spacy
ℹ Saving to output directory: output
ℹ Using CPU

=========================== Initializing pipeline ===========================
[2021-12-05 18:37:25,938] [INFO] Set up nlp object from config
[2021-12-05 18:37:25,963] [INFO] Pipeline: ['tok2vec', 'ner']
[2021-12-05 18:37:25,966] [INFO] Created vocabulary
[2021-12-05 18:37:25,967] [INFO] Finished initializing nlp object
[2021-12-05 18:37:27,839] [INFO] Initialized pipeline components: ['tok2vec', 'ner']
✔ Initialized pipeline

============================= Training pipeline =============================
ℹ Pipeline: ['tok2vec', 'ner']
ℹ Initial learn rate: 0.0
E    #       LOSS TOK2VEC  LOSS NER  ENTS_F  ENTS_P  ENTS_R  SCORE
---  ------  ------------  --------  ------  ------  ------  ------
  0       0          0.00     97.06    0.00    0.00    0.00    0.00
  5     200         24.47  20126.19    0.00    0.00    0.00    0.00
 13     400         60.80   4871.57    4.37   17.24    2.50    0.04
 22     600         58.02   3135.43   19.92   16.78   24.50    0.20
 32     800         68.49   2327.81   19.48   15.57   26.00    0.19
 45    1000         63.39   1839.45   18.49   13.92   27.50    0.18
 61    1200         55.58   1247.44   17.17   12.59   27.00    0.17
 81    1400         45.46    697.75   15.64   11.28   25.50    0.16
104    1600         41.39    357.29   14.79   10.34   26.00    0.15
133    1800         27.55    120.69   14.76   10.23   26.50    0.15
168    2000         18.28     48.14   15.80   11.00   28.00    0.16
209    2200         10.91     16.57   15.99   11.11   28.50    0.16
✔ Saved pipeline to output directory
output/model-last
```

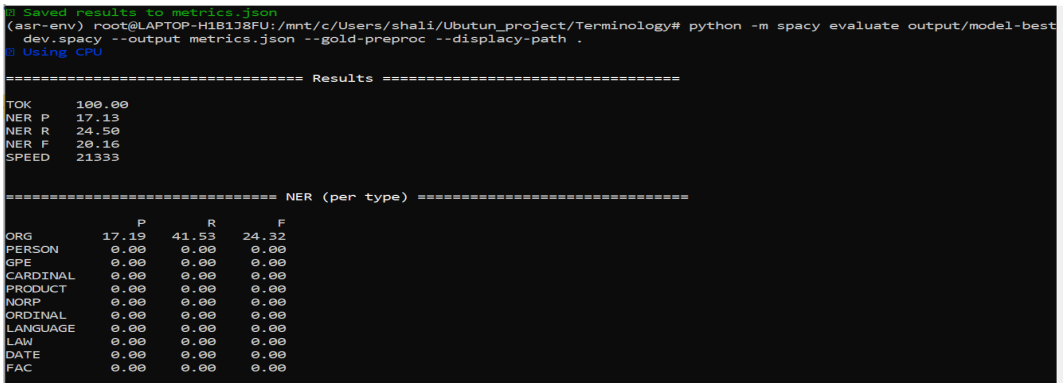
   (b) Evaluation 1 :Evaluation was carried out on train.spacy (annotated iob corpus) and it was able to train based on the keywords with following scores



```
(asr-env) root@LAPTOP-H1B1J8FU:/mnt/c/Users/shali/Ubutun_project/Terminology# python -m spacy evaluate output/model-best
 train.spacy --output metrics.json --gold-preproc --displacy-path .
ℹ Using CPU

================================== Results ==================================

TOK     100.00
NER P   65.13
NER R   45.45
NER F   53.54
SPEED   9545


================================ NER (per type) ================================

                P       R       F
PERSON       0.00    0.00    0.00
GPE          0.00    0.00    0.00
DATE         0.00    0.00    0.00
ORG         65.13   83.78   73.29
CARDINAL     0.00    0.00    0.00
NORP         0.00    0.00    0.00
LANGUAGE     0.00    0.00    0.00
ORDINAL      0.00    0.00    0.00
```

Precision	Recall	F Score
65.13	45.45	53,54

   (c) Evaluation 2 :Evaluation was carried out on unseen article and it was able extract keywords but didn't perform so well with following scores



```
✔ Saved results to metrics.json
(asr-env) root@LAPTOP-H1B1J8FU:/mnt/c/Users/shali/Ubutun_project/Terminology# python -m spacy evaluate output/model-best
 dev.spacy --output metrics.json --gold-preproc --displacy-path .
ℹ Using CPU

================================== Results ==================================

TOK     100.00
NER P   17.13
NER R   24.50
NER F   20.16
SPEED   21333


================================ NER (per type) ================================

                P       R       F
ORG         17.19   41.53   24.32
PERSON       0.00    0.00    0.00
GPE          0.00    0.00    0.00
CARDINAL     0.00    0.00    0.00
PRODUCT      0.00    0.00    0.00
NORP         0.00    0.00    0.00
ORDINAL      0.00    0.00    0.00
LANGUAGE     0.00    0.00    0.00
LAW          0.00    0.00    0.00
DATE         0.00    0.00    0.00
FAC          0.00    0.00    0.00
```

Precision	Recall	F Score
17.13	24.50	20.16

3 Challenges

- Manually extraction of keywords is a tedious work which took lot of time to filter out the actual keywords
- Adding less constrains in the rule-based system would filter out some exact terms from the text, so that we couldn't include those as terms.
- Accuracy in the results. We need to test the model using different pipelines to get a better result. We also believe that the amount of data used for training model will play a role in producing a better result; we think that our model would perform much better than this if there were a big set of annotated corpora used for training the model.

4 Conclusions

After completing this project, we can have a better understanding of how to extract terms in any domain by using rule-based system and neural model along with different term extraction methods, such as phrase pattern matching, statistical approach, and manual annotation. We have proposed and implemented an automatic keyword extraction system using Spacy model, and its performance was evaluated. The model has not given satisfactory performance as our expectations, but there is a scope for improvement using different pipelines.The experimental results show that a Spacy model can perform considerably well in the task of sequence labelling.The extracted keywords are able to effectively convey a short description about the content of the article;however, with the help of better lemmatizers and parsers,the performance of the model can be improved considerably.